Introduction to STATA version 11

What is Stata?

Stata is a powerful statistical software package that provides a wide range of basic and advanced data analysis capabilities.

1. Introduction

(a) It is much easier to use a computer package once you have used it! Hence this practical deliberately introduces you to the package Stata before you know much about it.

(b) Be adventurous. If you are not sure whether something is correct, try it out.

(c) Stata has a large number of commands, e.g. **summarize**. Most commands can either be accessed via the menu system or they can be typed into, and run from the Command window.

2. Starting Off

Go into Stata by **double-clicking** on the Stata icon. In Stata you view a file or work on it in a window. Four windows will open automatically on starting; these are:

(a)	**Results**	- for displaying output, except high resolution graphics.
(b)	**Command**	- for typing and running commands.
(c)	**Review**	**-** for displaying past commands.
(d)	**Variables**	- for listing the variables in your data.

Note some windows may be hidden from view. To see them rearrange the visible windows.

A type of window which does not open automatically is a **Graph** window. This window is used to display high resolution graphical output.

3. A Simple Example

Let us enter a small data set into the worksheet and carry out some simple analyses. There are three columns of data to enter, with a variable name (coding) for each column (i.e. x1, x2, and x3).

x1	**x2**	**x3**
10	15	20
12	15	19
22	11	26
41	9	22

To enter the data open the Data Editor window use **Window → Data Editor**. It is basically a spreadsheet like system, displaying a worksheet.

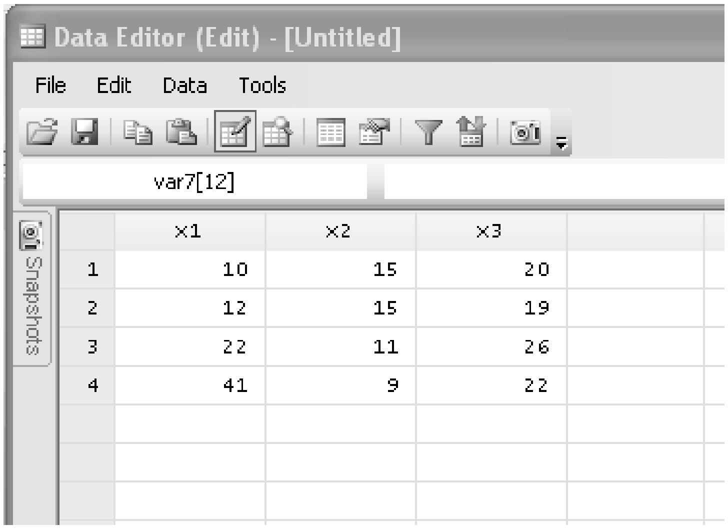
Enter the data directly into the Data Editor window, typing the numbers into the first four rows labelled **1** to **4** of the first three columns. After entering each number press the **Enter** key (↵) on the keyboard.

Stata automatically numbers columns (variables) in the worksheet as **var1**, **var2**, **var3**, ... etc. We will assign the names **x1, x2,** and **x3** to columns **var1** to **var3** respectively.

- **Right-click** inside the column/variable named **var1** and in the **Variable Properties** dialog box rename the variable **x1**. **Click Apply**

- In the same way name the other two variables

- Your Data Editor window should now look like the one below



var7[12]

	x1	x2	x3
1	10	15	20
2	12	15	19
3	22	11	26
4	41	9	22

Before you can continue you need to close the Data Editor window, by **clicking** on ☒ in the top right-hand corner.

Variable names can contain up to 32 characters, consisting of A-Z, a-z, 0-9 and _.
There must be no spaces or other characters.
Names are case sensitive, so *COUNT* is different from *count*, and must start with a letter or _.

Next we will produce a scatter plot of **x2** vs. **x1**.
Move to the **Command** window and type:

```
. scatter x2 x1
```

then press Enter.

You will now see the scatter plot in its own Graph window. This particular graph is not very informative but neither are our data. To discard the plot **click** on ☒ in the top right-hand corner of the Graph window.

Next, create a line plot with the command:

```
. line x2 x1
```

and press **Enter**.

Graphs can also be edited and saved as separate files which can be included in word processed reports (use **File → Save** from within the Graph window, for example).

Every time you create a graph, the new chart overwrites the existing chart. To keep a chart, use the name option as follows:

```
. scatter x2 x1, name(scatter)

. line x2 x3, name(line)
```

Now the two graphic windows are visible at the same time. You can even combine these two graphics into a single Graphic window with:

```
. graph combine scatter line
```

Note that a list of the commands used, either directly or via dialog boxes, is given in the Review window.

Any line can be transferred to the Command window by **clicking** on it.

Command lines can also be recalled into the Command window by pressing the **PageUp** key repeatedly.

4. Descriptive Statistics

In the Command window type

```
. summarize x1 x2
```

and execute it by pressing Enter.

This command prints a number of descriptive statistics including the sample size (Obs), mean, standard deviation (Std. Dev.), minimum, and maximum for the two variables.

A more flexible command is –tabstat- which allows the user to choose many more summaries.
See which statistics are available with:

```
. help tabstat
```

which opens an on-line help file for this very command.

Then try:

```
. tabstat x1 x2, stat(n mean sd q iqr)
```

try one more option now, with:

```
. tabstat x1 x2, stat(n mean sd q iqr) col(stats)
```

5. Saving

5.1 Saving data

Before moving to the next example we will save the data currently stored in the Data Editor.

Change the current working directory to c:\user with

```
. cd c:\user
```

now save the current dataset in c:user with

```
. save first
```

by default the data is saved in Stata format with the extension ***.dta**, as stressed by the message.

5. 2 Saving analysis output

To save output in the Results window you need to create a log file BEFORE you perform any analysis. The results are then sent to this file, which can be printed and viewed later.

To begin recording an analysis session, specify the file name as **first** which by default is stored on c:\user. Use the –log using- command as follows:

```
. log using first
```

A message is printed in the Results window, saying you have started sending output to a log file.

By default the file is saved as a Stata formatted file with extension *.**scml**.

Now do something that produces results in the **Results** window, for example produce some summary statistics.

Now stop sending output to the log file with

```
. log close
```

The output saved to **analysis.smcl** can now be viewed by issuing the following command:

```
. view first.smcl
```

The contents are now displayed in the **Viewer**, another type of Stata window. This window is for viewing Help files, searching, viewing log files, and more. Close the **Viewer** when you have finished.

The format of the output file can be changed to ASCII format by using the –text- option of the –log- command as follows:

```
. log using first, text
```

6. Installing user-written packages

We will need to use a third-party contributed function – xi3 – later, which is not included with the base installation of Stata. Install it as follows:

```
. net install xi3
```

7. Opening Stata Data Files

Let us practice opening a saved Stata data file. Firstly, clear the Data Editor with

```
. clear
```

Now open the file you just saved, with:

```
. use first
```

Check the data with

```
. list
```

then try

```
. desc
```

Note that Stata provides a comprehensive on-line help, which is orientated towards using the command language. So for example, you could try:

```
. help desc
```

8. Calculator

Stata can be used interactively as a calculator.

Calculate two new columns as follows:

```
. generate x1x2 = x1*x2
. generate x1_sq = x1^2
```

Note that – is used for subtraction, + for addition, * for multiplication, / for division, ^ for raising a number to a power.

Stata also has a range of functions. For example, to calculate the square root of x1, try

```
. gen sqrt_x1 = srqt(x1)
```

Statistical functions are available: to calculate and save the mean of x1 in a new column, use

```
. egen mean_x1 = mean(x1)
```

Now look at the data file with

```
. list
```